College Football Statistics Project

Cody Faircloth

Who Am I?

- Name: Cody Faircloth

- Age: 25 years old

- Occupation: Data Engineer

- Education: BS Computer Science, minor in Software Engineering

Why Tech?

- Childhood Interest

- Constraints Choosing a College Major

- The Process and Continuous Learning

Technologies Used

- Programming Language
  - Python

- Relational Database Management System (RDBMS)
  - MySQL

- Containerization Platform
  - Docker

- Docker Images
  - Alpine
  - Apache Airflow
  - Apache Spark

- Python Libraries
  - Airflow
  - Dash
  - DynaConf
  - SQLAlchemy
  - SSHOperator
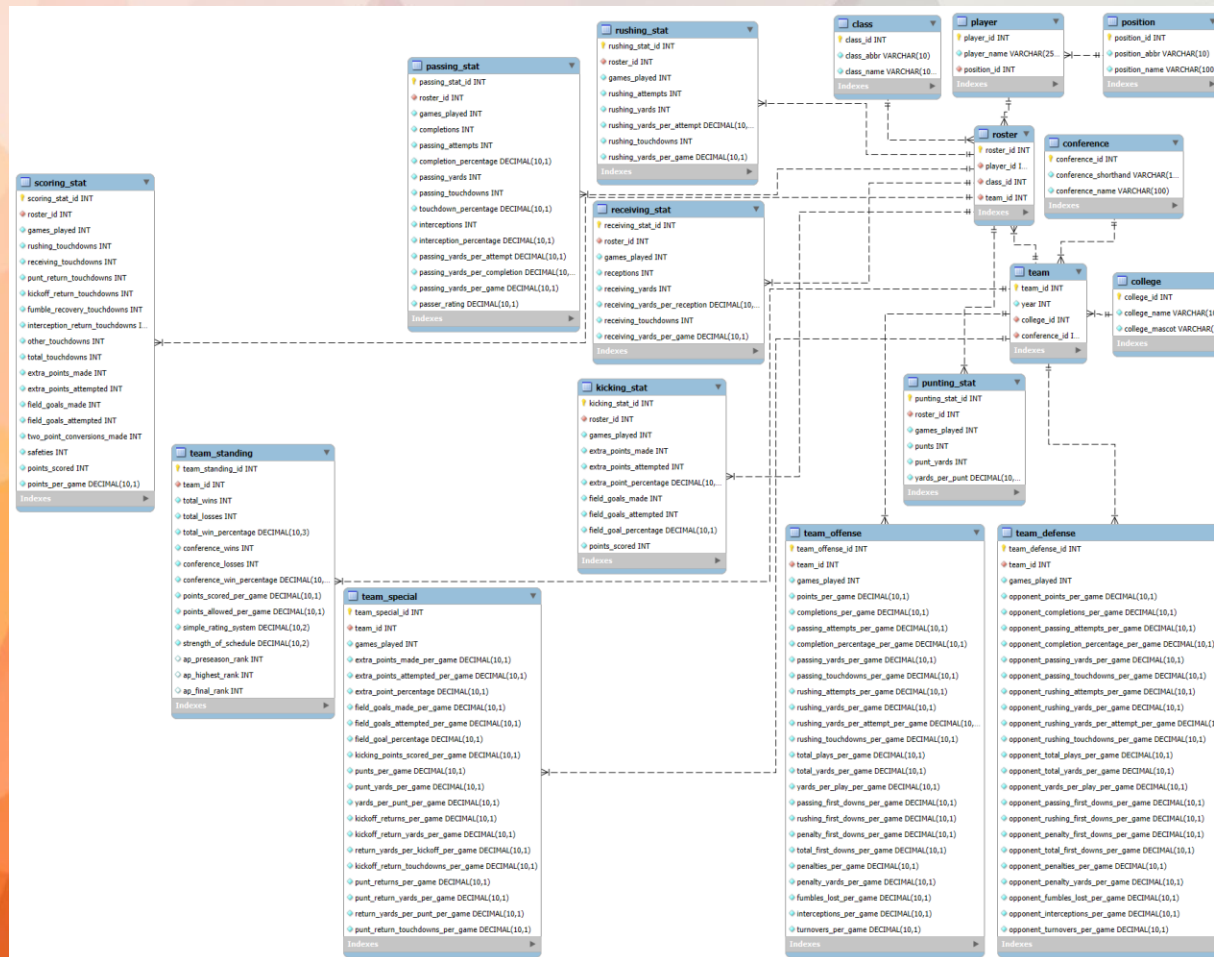 - Pandas
  - Pendulum
  - Plotly Express
  - PySpark

Goals

- Scalable Dashboard Similar to ESPN

- Identify Trends by Team, Conference, and Nationally

Schema Design Process

- Complex Dataset

- Needs Scalability

- Complex Schema

- Practical Normalization

Finalized Schema

ETL

- Scalable

- Separation of Concerns

- Relatively Efficient

```python
# Functions to build dataframes for ETL process
> def build_master_college_conf(years): ...

> def build_master_players(year, teams): ...

> def build_conferences(master): ...

> def build_colleges(master): ...

> def build_teams(master, colleges, conferences): ...

> def build_positions(master): ...

> def build_classes(master): ...

> def build_players(master, positions): ...

> def build_rosters(master, players, teams, classes): ...

> def build_rushing(years,rosters): ...

> def build_receiving(years, rosters): ...

> def build_passing(years, rosters): ...

> def build_scoring(years, rosters): ...

> def build_punting(years, rosters): ...

> def build_kicking(years, rosters): ...

> def build_team_offense(years, teams): ...

> def build_team_defense(years, teams): ...

> def build_team_special(years, teams): ...

> def build_standings(years, teams): ...

# List of years to process
years = [2024, 2023, 2022, 2021, 2020]
```

Airflow

- Schedule ETL Process

- Run Schema SQL Script

- Run Spark command

```python
import pendulum
from airflow.decorators import dag
from airflow.providers.common.sql.operators.sql import SQLExecuteQueryOperator
from airflow.providers.ssh.operators.ssh import SSHOperator

# Create a DAG to run cfb_statistics_schmema.sql and cfb_stats_app.py
@dag(
    schedule_interval=None,
    start_date=pendulum.datetime(2025, 1, 1, tz="UTC"),
    catchup=False,
    tags=["cfb_stats", "ssh", "mysql"],
)
def ssh_mysql_cfb_stats_dag():
    # Define the SSH command to run cfb_stats_app.py
    command = """
    export JAVA_HOME=/opt/java/openjdk;
    /opt/spark/bin/spark-submit --master spark://spark-leader-cfb:7077 /opt/spark/work-dir/
    cfb_stats_app.py
    """
    # Create SSHOperator task to run cfb_stats_app.py
    ssh_cfb_stats = SSHOperator(
        ssh_conn_id="ssh_leader", task_id="ssh_task", command=command, cmd_timeout=600
    )
    # Create SQLExecuteQueryOperator task to run cfb_statistics_schema.sql
    cfb_stats_schema = SQLExecuteQueryOperator(
        task_id="cfb_stats_schema",
        sql="/cfb_statistics_schema.sql",
        conn_id="mysql_sys"
    )

    # Schedule tasks
    cfb_stats_schema >> ssh_cfb_stats

ssh_mysql_cfb_stats_dag()
```

Visualizations

- Roster and Stat Tables

- Team Standings in Conference and Nationally

- Stats over Time Graphs

- Stat Distribution

Visualization Examples

Stats by Team: Roster and Header

Stats by Team: Team Stats Tab

Visualization Examples (cont.)

Stats by Conference:
Standings and Header

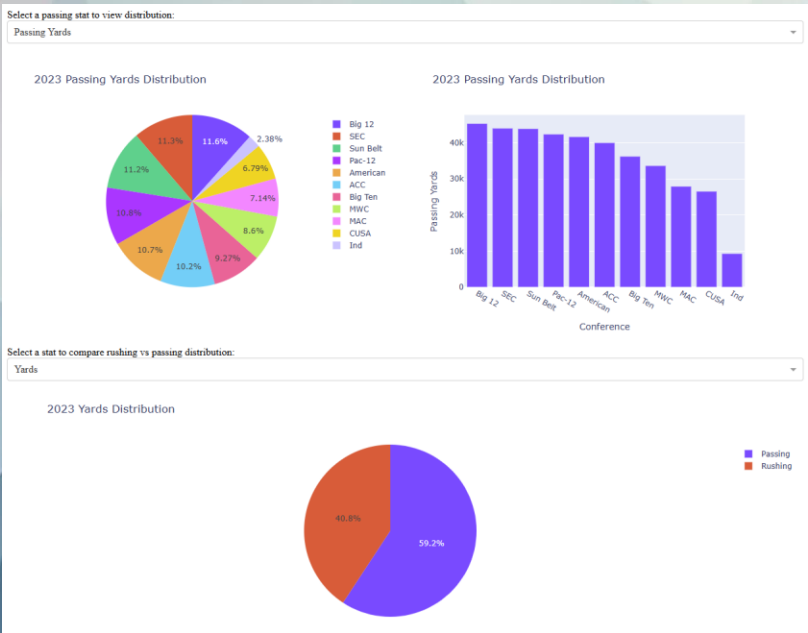Stats by Conference:
Individual Stats

Visualization Examples (cont.)
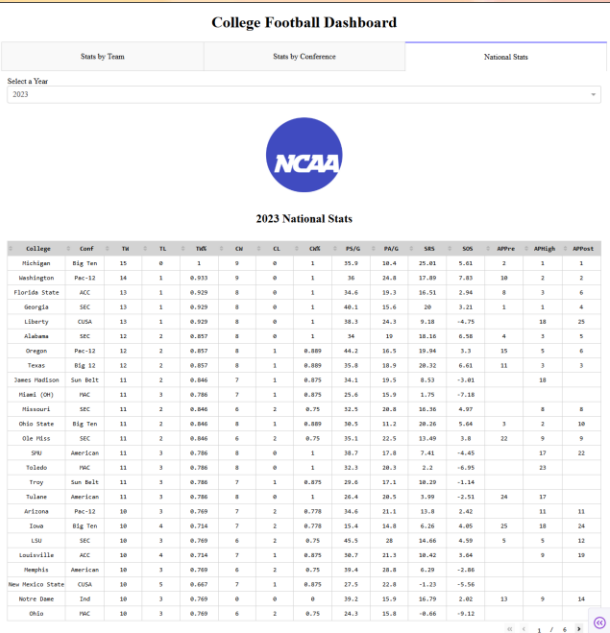
National Stats: Standings and Header

Stats by Team: Team Stats over Time

National Stats: Conference Stat Distribution

Conclusions

- Dashboard displays accurate data

- Dashboard data retrieval is efficient

- Trends Conclusions vary by team

- Distribution of play types favor passing